

UDC 004.6

SCOPUS CODE 1701

<https://doi.org/10.36073/1512-0996-2021-4-37-45>

Covid-19-თან დაკავშირებული შეტყობინებების ანალიზი და ვიზუალიზაცია 112-ში

რუსუდან ქუთათელაძე	ბიზნესის ადმინისტრირების დეპარტამენტი, საქართველოს ტექნიკური უნივერსიტეტი, საქართველო, 0160, თბილისი, მ. კოსტავას 77 E-mail: r.kutateladze@gtu.ge	ტექნიკური
ანა კობიაშვილი	ეკონომიკური ინფორმატიკის დეპარტამენტი, საქართველოს უნივერსიტეტი, საქართველო, 0160, თბილისი, მ. კოსტავას 77 E-mail: a.kobiashvili@gtu.ge	ტექნიკური
ნოდარ დარჩიაშვილი	ეკონომიკური ინფორმატიკის დეპარტამენტი, საქართველოს უნივერსიტეტი, საქართველო, 0160, თბილისი, მ. კოსტავას 77 E-mail: nodo619@gmail.com	ტექნიკური

რეცენზენტები:

ე. ბარათაშვილი, ბიზნესტექნოლოგიების ფაკულტეტის პროფესორი

E-mail: e.baratashvili@gtu.ge

მ. კიკნაძე, ინფორმატიკისა და მართვის სისტემების ფაკულტეტის პროფესორი

E-mail: m.kiknadze@gtu.ge

ანოტაცია. კორონავირუსის პანდემიამ მნიშვნელოვნად შეცვალა ადამიანების ცხოვრება და მათი საქმიანობის თითქმის ყველა სფერო. ერთ-ერთი ორგანიზაცია, რომლის ფუნქციონირება პირდაპირაა დაკავშირებული პანდემიის მართვასთან, არის გადაუდებელი დახმარების ოპერატიული მართვის ცენტრი 112. ის აქტიურად არის ჩართული კორონავირუსის მართვის პროცესში და უკვე წელიწადზე მეტია, რაც განსაკუთრებულ რეჟიმში მუშაობს, რა-

თა თითოეულმა მოქალაქემ დროული და მაღალკვალიფიციური მომსახურება მიიღოს.

სტატიაში აღწერილია გადაუდებელი დახმარების ოპერატიული მართვის ცენტრ 112-ში დამუშავებული კოვიდ-19-ის შემთხვევების მონიტორინგის სისტემა, რომლის რეალურ დროში სტაბილურ მუშაობას პანდემიის პირობებში გაზრდილმა შემოსულმა ზარებმა საფრთხე შეუქმნა. აქედან გამომდინარე, შეიქმნა ახალი ფუნქციონალი, რომლის დანიშნულებაცაა სასწრაფო დახმარების ბრიგადების ოპტიმალური ტერიტორიული განაწილება. სტა-

ტიაში განხილულია აღნიშნული ფუნქციონალის Front end ნაწილი. აღწერილია აპლიკაციაში რუკების ინტეგრირების პროცესი, რის შედეგადაც ხდება სასურველი მონაცემების რუკის ფენებად გარდაქმნა და ვიზუალურად წარმოდგენა. აპლიკაცია გვიჩვენებს კოვიდის შემთხვევების რაოდენობას დამატებით ტექსტთან ერთად. ნაჩვენებია კოვიდის შემთხვევების რუკის მაგალითი. მოცემულია აპლიკაციის მუშაობის ილუსტრაციები, რომლებიც გვიჩვენებს როგორც მიმდინარე დღის, ისე გასული პერიოდის სტატისტიკურ მონაცემებს.

საკვანძო სიტყვები: Google-ის რუკა; Front end აპლიკაცია; მონაცემთა დამუშავება; სატელეფონო ცენტრი; open source ბიბლიოთეკა; Angular ტექნოლოგია.

შესავალი

პანდემიის პერიოდში 112-ში დღე-ღამის განმავლობაში განხორციელებული ზარების რაოდენობა დაახლოებით 1,4-ჯერ გაიზარდა. ამასთანავე, კვლევების თანახმად, მიუხედავად მძიმე ეპიდემიოლოგიისა, 112-ში დღე-ღამის განმავლობაში 3000-მდე არამიზნობრივი და საინფორმაციო ხასიათის ზარი შედის, რაც ხელს უშლის მუშაობის ნორმალურად წარმართვას. შედეგად კი მიიღება უკმაყოფილო აბონენტი, რომელსაც შესაძლოა რეალური სასწრაფო დახმარება სჭირდება, მაგრამ გადატვირთულობის გამო სატელეფონო ცენტრს ვერ უკავშირდება.

ინფიცირების რაოდენობის ზრდასთან ერთად, იზრდება ცენტრის დატვირთვა. 112 განაგრძობს 24 საათის განმავლობაში განსაკუთრებულ რეჟიმში მუშაობას, რათა თითოეულმა მოქალაქემ მაღალი სტანდარტის მომსახურება მიიღოს, როგორც გადაუდებელი შემთხვევის დროს, ასევე ვირუსთან დაკავშირებით [1].

ძირითადი ნაწილი

112-ის ოპერატორი იღებს ინფორმაციას, დაამუშავებს, შეაფასებს მას და გადაუზარუნებს შესაბამის სამსახურს შემდგომი რეაგირებისათვის. მიუხედავად იმისა, რომ პანდემიის მეორე ტალღის პიკის პერიოდში ოპერატორების რაოდენობა თითქმის 3-ჯერ გაიზარდა, სამსახურის გამართული მუშაობისათვის საჭირო გახდა სპეციალური პროგრამული უზრუნველყოფის დამუშავება, რომლის მეშვეობითაც ხდება სასწრაფო დახმარების ბრიგადების ოპტიმალური გადანაწილება საქართველოს რეგიონებსა და თბილისის სხვადასხვა რაიონში.

იმის გამო, რომ 112-ში განხორციელებული ზარი მხოლოდ კორონავირუსის შემთხვევებს არ ეხება, სასწრაფო დახმარების ბრიგადების ეფექტური მუშაობისათვის მნიშვნელოვანია კოვიდ-19-თან დაკავშირებული გამოძახებების სტატისტიკური ანალიზი. ამიტომ 112-ის მონაცემთა მონიტორინგის სისტემას დაემატა ახალი ფუნქციონალი, რომელიც გვიჩვენებს კორონავირუსთან დაკავშირებული ზარების სტატისტიკას რეგიონული და რაიონული კატეგორიზაციით [2]. სტატისტიკური მონაცემები ორი სახით მიიღება: ცხრილის სახით და რუკაზე ვიზუალიზაციით.

დამუშავებული აპლიკაციის მთავარი უპირატე- სობაა მონაცემთა ბაზაში მიმდინარე მონაცემების ავტომატური განახლების შესაძლებლობა. პროგ- რამა ყოველი იტერაციის შესრულებისას აღმოაჩენს, არსებობს თუ არა განახლებული მონაცემები და შესაბამის მონაცემებს აგზავნის Front end აპლიკა- ციასთან.

სისტემის Front end ნაწილი არის აპლიკაცია, რომელიც შესრულებულია Angular ტექნოლოგიის გამოყენებით [3]. აპლიკაციაში ინტეგრირებულია რუკები Leaflet ბიბლიოთეკის გამოყენებით [4]. მო- ნაცემთა რეალურ დროში გასაახლებლად პროგრამა იყენებს SignalR შეტყობინებების სისტემას [5].

Leaflet არის JavaScript-ის open source ბიბლიო- თეკა, რომელიც Front end აპლიკაციაში რუკების ინტეგრირების საშუალებას გვაძლევს [6]. Leaflet-ის გამოყენებით შესაძლებელია სასურველი მონაცემე- ბის რუკის ფენებად გარდაქმნა და ვიზუალური წარმოდგენა.

აღსანიშნავია, რომ Leaflet არის მხოლოდ ინს- ტრუმენტი და არა უშუალოდ რუკა. მისი გამოყენე- ბისთვის აუცილებელია მივაწოდოთ როგორც რუ- კის გრაფიკული გამოსახულება, ისე მარკერების მონაცემები.

კოვიდ-19-ის აპლიკაცია Leaflet-ის ვიზუალურ წყაროდ იყენებს Google-ის რუკას [7] (სურ.1). ობიექტის Typescript აღწერა ასეთია:

```
googleMap =
L.tileLayer('https://{s}.google.com/vt/lyrs=m&x=
{x}&y={y}&z={z}', {
maxZoom: 21,
minZoom: 8,
```

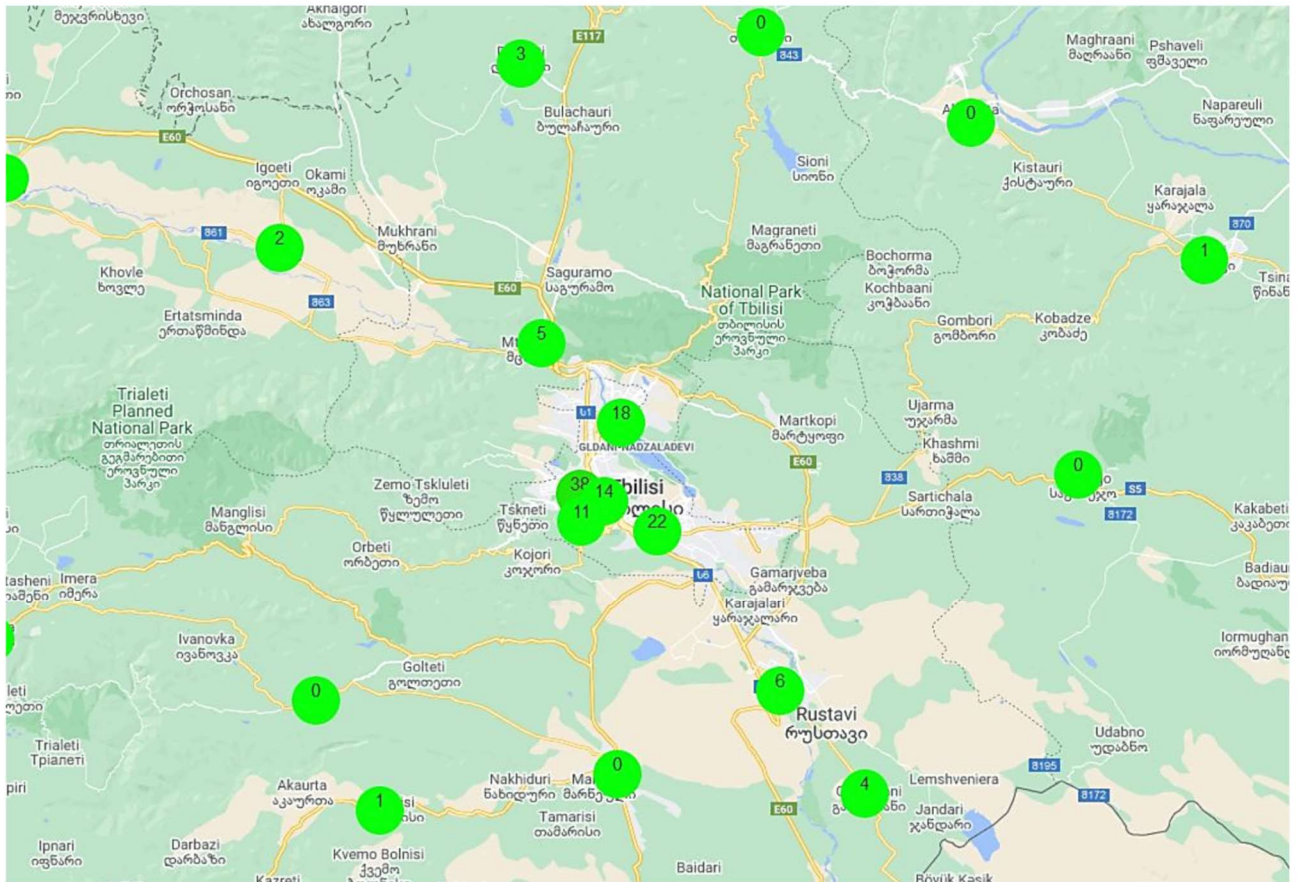
```
subdomains: ['mt0', 'mt1', 'mt2', 'mt3']
});
```

tileLayer ობიექტი გამოიყენება რუკის ჩასატვირ- თად. ის იღებს შემდეგ პარამეტრებს:

- რუკის URL მისამართი. ამ შემთხვევაში პარამეტ- რად ვუთითებთ მისამართს Google-ის რუკაზე. ეს მნიშვნელობაც, თავის მხრივ, შეიცავს შიგა ტექსტურ პარამეტრებს: s, x, y და z.
- s აღნიშნავს რუკის ალტერნატიულ ქვედომენებს და საშუალებას აძლევს Leaflet აპლიკაციას პარა- ლელური მოთხოვნები გაგზავნოს რამდენიმე ქვედომენზე.
- x და y: მარკერის კოორდინატები.
- z: მიახლოების მნიშვნელობა რუკის ჩატვირთ- ვისას.
- maxZoom: რუკის მიახლოების მაქსიმალური მნიშვნელობა.
- minZoom: რუკის მიახლოების მინიმალური მნიშვნელობა.
- subdomains: ზემოთ აღწერილი ქვედომენის შე- საძლო მნიშვნელობების ჩამონათვალი. ეს მნი- შვნელობები მოთავსდება s პარამეტრის ადგი- ლას.

რუკაზე რაიონის მარკერის მოთავსების კოდი:

```
const addCoronaStatisticsToMap = (name, data) => {
const percent = (data.coronaStatisticInfo.casesCount
/ this.totalCoronaCases) * 100;
const roundedPercent = Math.round(percent);
const marker = L.marker ([data. coordinatesInfo.
longitude, data.coordinatesInfo.latitude], {
icon: L.divIcon({
html: ` ${data.coronaStatisticInfo.casesCount} ` ,
```



სურ. 1. კოვიდ-19-ის შემთხვევების რუკის მაგალითი

```

className:
` ${this.appService.getCoronaStatisticColors
(roundedPercent)} `,
iconSize: L.point(40, 40)
}),
title: ` ${name} ` });
this.coronaStatisticsOnMap[name] = {data, marker };
this.coronaStatisticsLayer.addLayer(marker);
bindPopupToCoronaStatistics(name);
};

```

addCoronaStatisticsToMap ფუნქციას აქვს ორი პარამეტრი: name – რაიონის სახელი, data – მონაცემები შემთხვევების რაოდენობის შესახებ. percent ცვლადში გამოითვლება მოცემული რაიონის პროცენტული მაჩვენებელი, ყველა რაიონის შემთხვევებიდან, ხოლო roundedPercent არის ამ მაჩვენებლის დამრგვალებული მნიშვნელობა. რუკაზე მარკერის ფენის შესაქმნელად ვიყენებთ Leaflet-ის marker ფუნქციას L.marker(): პირველი პარამეტრი: გრძედისა და განედის მასივი.

Icon: მარკერის ვიზუალური გამოსახულების აღწერა.

html: მარკერის HTML აღწერა.

className: CSS კლასის სახელი, რომელიც გვსურს, რომ მიენიჭოს მოცემულ მარკერს.

iconSize: მარკერის ზომა.

title: მარკერის tooltip-ის მნიშვნელობა.

რუკაზე რაიონის დამატებისას ობიექტს ვინახავთ ლექსიკონის ტიპის ობიექტში (dictionary), რათა შევძლოთ მისი განახლება.

```
this.coronaStatisticsOnMap[name] = { data, marker };
```

coronaStatisticsLayer არის Leaflet-ის FeatureGroup ტიპის ობიექტი, რომლის addLayer ფუნქციით ახალშექმნილ მარკერსაც ვამატებთ რუკაზე. this.coronaStatisticsLayer.addLayer(marker);

Leaflet საშუალებას გვაძლევს შევქმნათ Popup კონტეინერი და მასში მოვათავსოთ მონაცემები სასურველი ფორმით. Popup-ის ჩვენება შეგვიძლია ნებისმიერი მოქმედების შესრულებისას. ამ აპლიკაციის შემთხვევაში, popup-ს ვაჩვენებთ მარკერზე კურსორის დაწკაპუნებისას.

Popup-ის შექმნის ფუნქცია:

```
const bindPopupToCoronaStatistics = name => {
  const { data, marker } = this.coronaStatistics On Map
  [name];
```

```
  let content = `<dl>
  <dt>რაიონი</dt>
  <dd>${data.districtName}</dd>
  `;
  if (data.pastCoronaStatisticInfo !== null) {
    content += `
  <dt>ბოლო 24 საათის სტატისტიკა</dt>
  <dd>${data.coronaStatisticInfo.casesCount}</dd>
```

```
<dt>წინა 24 საათის სტატისტიკა</dt>
```

```
<dd>${data.pastCoronaStatisticInfo.casesCount}</dd>
```

```
`;
```

```
} else {
```

```
  content += `
```

```
<dt>პერიოდის სტატისტიკა</dt>
```

```
<dd>${data.coronaStatisticInfo.casesCount}</dd>
```

```
`;
```

```
}
```

```
content += '</dl>';
```

```
marker.bindPopup(content);
```

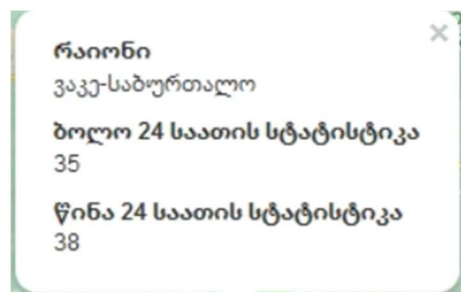
```
};
```

bindPopupToCoronaStatistics პარამეტრად იღებს რაიონის სახელს და მისი გამოყენებით იღებს აუცილებელ მონაცემებს coronaStatisticsOnMap ლექსიკონის ობიექტიდან: const { data, marker } = this.coronaStatisticsOnMap[name];

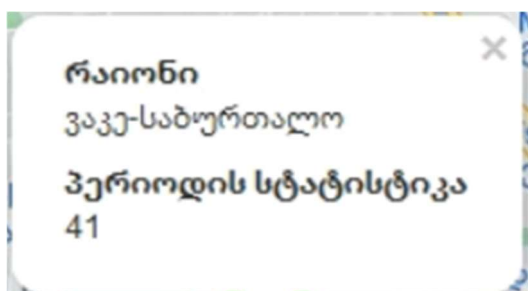
content ცვლადი აღნიშნავს popup კონტეინერის HTML კოდს და გვიჩვენებს კოვიდის შემთხვევების რაოდენობას დამატებით ტექსტთან ერთად.

marker.bindPopup(content) ფუნქცია ახალშექმნილ content-ს მიაბამს მარკერს.

Popup კონტეინერის ვიზუალური მხარე მოცემულია მე-2 და მე-3 სურ-ებზე.



სურ. 2. მიმდინარე დღის მონაცემები წინა დღის მონაცემებთან ერთად



სურ. 3. გასული პერიოდის მონაცემები (კონკრეტული დღეების მითითებით)

სისტემაში ასევე შესაძლებელია მონაცემების განახლება რეალურ დროში. SignalR არის ASP.NET ბიბლიოთეკა, რომელიც იყენებს არსებულ მონაცემთა გადამცემ ტექნოლოგიებს არჩეული ინფრასტრუქტურის გათვალისწინებით. SignalR-ს აქვს რეალურ დროში კომუნიკაციის შესაძლებლობა ფართო დიაპაზონის აპლიკაციებისათვის. ამრიგად, დეველოპერს არ სჭირდება იფიქროს ტრანსპორტირების მექანიზმზე და იმაზე, თუ რა ტექნოლოგია გამოიყენოს იმ შემთხვევაში, როდესაც მის ინფრასტრუქტურას რომელიმე ტექნოლოგიის მხარდაჭერა არა აქვს.

SignalR-ის საშუალებით Back end სერვისი, მონაცემების განახლებისას, შეტყობინებას უგზავნის Front end აპლიკაციას, რომ რუკაზე ამა თუ იმ მარკერის განახლებაა საჭირო.

სერვისისგან მიღებული შეტყობინების დამუშავების კოდი:

```
this.coronaStatisticsHubConnection.on('StatisticsUpdated', this.updateCoronaStatisticsState.bind(this));
```

მოცემული კოდი Back end-სგან მიღებული 'StatisticsUpdated' შეტყობინების საფუძველზე ასრულებს updateCoronaStatistics ფუნქციას. ფუნქციას გადაეცემა შემდეგი პარამეტრები:

districtId: იმ რაიონის Id, რომლის მნიშვნელობაც განახლდა.

statisticsInfo: განახლებული მნიშვნელობა.

მონაცემების ვიზუალური ასახვისთვის update CoronaStatistics ფუნქცია მიმართავს update CoronaStatisticsOnMap-ს:

```
const updateCoronaStatisticsOnMap = (name, newData) => {
    const marker =
this.coronaStatisticsOnMap[name].marker;
    const percent =
(newData.coronaStatisticInfo.casesCount /
this.totalCoronaCases) * 100;
    const roundedPercent = Math.round(percent);
    marker.setIcon(L.divIcon({
    html: ` ${newData.coronaStatisticInfo.casesCount} `,
    className:
` ${this.appService.getCoronaStatisticColors(roundedPercent)} `,
    iconSize: L.point(40, 40)
}))
    const newCoordinates = newData.coordinatesInfo;
    this.coronaStatisticsOnMap[name].data = newData;
    marker.setLatLng([newCoordinates.longitude,
newCoordinates.latitude ]);
    bindPopupToCoronaStatistics(name);
};
```

marker.setIcon – მარკერის ვიზუალური ფორმის განახლება.

marker.setLatLng ბრძანება განახლებს კოორდინატებს არსებულ marker-ში.

ფუნქციის დანარჩენი ნაწილი იგივეა, რაც მარკერის შექმნის ფუნქციაში, იმ განსხვავებით, რომ აქ

მონაცემთა ცვლადს newData სახელით მივმართავთ.

დასკვნა

გადაუდებელი დახმარების ოპერატიული მართვის ცენტრის მუშაობის ეფექტურობის ასამაღლებლად შექმნილი ფუნქციონალი საშუალებას იძლევა სისტემაში მონაცემები განახლდეს რეალურ დროში. აღნიშნული აპლიკაცია ეფუძნება ASP.NET ბიბლიოთეკას და იყენებს არსებული მონაცემების გა-

დაცემის ტექნოლოგიებს არჩეული ინფრასტრუქტურის გათვალისწინებით. აპლიკაცია იყენებს SignalR შეტყობინებების სისტემას, რომელსაც აქვს რეალურ დროში კომუნიკაციის შესაძლებლობა ფართო დიაპაზონის აპლიკაციებისათვის. აქედან გამომდინარე, დეველოპერს აღარ დასჭირდება იფიქროს ტრანსპორტირების მექანიზმზე და იმაზე, თუ რა ტექნოლოგია გამოიყენოს იმ შემთხვევაში, თუ მის ინფრასტრუქტურას არ ექნება რომელიმე ტექნოლოგიის მხარდაჭერა.

ლიტერატურა

1. Statement of the Public Safety Management Center of the Ministry of Internal Affairs, 112. (2020). Retrieved from: <https://police.ge/ge/shinagan-saqmeta-saministros-sazogadoebrivi-usaftrkxhoebis-martvis-tsentr-is-gantskhadeba/14058> (In Georgian);
2. Darchiashvili, N., Kobiashvili, A. Covid-19 Case Monitoring System at the Public Safety Management Centre. Georgian Technical University, Works, Automated Management Systems, Tbilisi, 2021, N2 (33), p. 54-61 (in Georgian);
3. JSON. (n.d). Introducing JSON. Retrieved from: <https://www.json.org/json-en.html>
4. Leaflet. (n.d.). *Leaflet — an open-source JavaScript library for interactive maps*. Leafletjs.Com. <https://leafletjs.com/reference-1.7.1.html>;
5. B. (2020, September 10). *Introduction to SignalR*. Microsoft Docs. Retrieved from: <https://docs.microsoft.com/en-us/aspnet/signalr/overview/getting-started/introduction-to-signalr>.
6. Microsoft. (n.d.). ASP.NET documentation. Microsoft Docs. Retrieved from: <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-5.0>;
7. Google. (n.d.). *Google Maps Platform Documentation*. Google Developers. <https://developers.google.com/maps/documentation>

UDC 004.6

SCOPUS CODE 1701

<https://doi.org/10.36073/1512-0996-2021-4-37-45>

Analysis and Visualization of Covid-19 Related Notifications Received at 112

- Rusudan Kutateladze** Department of Business Administration, Georgian Technical University, Georgia, 0160, Tbilisi, 77 M. Kostava str.
E-mail: r.kutateladze@gtu.ge
- Ana Kobiashvili** Department of Economic Informatics, Georgian Technical University, Georgia, 0160, Tbilisi, 77 M. Kostava str.
E-mail: a.kobiashvili@gtu.ge
- Nodar Darchiashvili** Department of Economic Informatics, Georgian Technical University, Georgia, 0160, Tbilisi, 77 M. Kostava str.
E-mail: nodo619@gmail.com

Reviewers:

E. Baratashvili, Professor, Faculty of Business Technologies, GTU

E-mail: e.baratashvili@gtu.ge

M. Kiknadze, Professor, Faculty of Informatics and Control Systems, GTU

E-mail: m.kiknadze@gtu.ge

Abstract. The coronavirus pandemic has significantly changed a person's life and almost all areas of his activity. One of the organizations whose activities are directly related to the management of the pandemic is the Emergency Management Center 112. It is actively involved in the process of managing the coronavirus and has been working in a special mode for more than a year to provide timely and highly qualified services to every citizen.

Covid-19 monitoring system developed at the 112 Emergency Management Center is described, whose real-time stability has been compromised by the increasing number of incoming calls during the pandemic. Consequently, a new functionality has been created, the purpose of which is the optimal geographical distribution of ambulance crews. The article discusses the front-end part of this functionality. It also describes the process of integrating maps into the application, which results in the transformation of the desired data into map layers and visual representation. The application shows the number of Covid-19 cases along with an additional text. An example of Covid-19 cases map is shown. The illustrations are provided showing the statistical data of both current days and past period.

Keywords: Angular technology; call center; data processing; front-end application; Google Map; open source library.

UDC 004.6

SCOPUS CODE 1701

HTTPS://DOI.ORG/10.36073/1512-0996-2021-4-37-45

Анализ и визуализация уведомлений, связанных с Covid-19, поступающих в 112

- Русудан Кутателадзе** Департамент бизнес-администрирования, Грузинский технический университет, Грузия, 0160, Тбилиси, ул. М. Костава, 77
E-mail: r.kutateladze@gtu.ge
- Анна Кобиашвили** Департамент экономической информатики, Грузинский технический университет, Грузия, 0160, Тбилиси, ул. М. Костава, 77
E-mail: a.kobiashvili@gtu.ge
- Нодар Дарчиашвили** Департамент экономической информатики, Грузинский технический университет, Грузия, 0160, Тбилиси, ул. М. Костава, 77
E-mail: nodo619@gmail.com

Рецензенты:

Е. Бараташвили, профессор факультета бизнес-технологий ГТУ

E-mail: e.baratashvili@gtu.ge

М. Кикнадзе, профессор факультета информатики и систем управления ГТУ

E-mail: m.kiknadze@gtu.ge

Аннотация. Пандемия коронавируса существенно изменила жизнь человека и практически все сферы его деятельности. Одной из организаций, деятельность которой напрямую связана с управлением пандемией, является Центр управления в чрезвычайных ситуациях 112. Он активно участвует в процессе управления коронавирусом и более года работает в специальном режиме для оказания своевременных и высококвалифицированных услуг каждому гражданину.

Описывается система мониторинга Ковид-19 случаев, разработанная в центре управления чрезвычайными ситуациями, в 112, стабильность работы в режиме реального времени которого оказалась под угрозой из-за увеличения количества входящих звонков во время пандемии. Следовательно, создан новый функционал, цель которого оптимальное территориальное распределение бригад скорой помощи. В статье обсуждается Front-end часть данного функционала. Описывается процесс интеграции карт в приложения, который приводит к преобразованию требуемых данных в слои карты и визуальное представление. Приложение показывает количество случаев Ковида вместе с дополнительным текстом. Показан пример карты Ковид случаев. Даны иллюстрации работы приложения, показывающие статистические данные как текущего дня, так и прошедшего периода.

Ключевые слова: колл-центр; библиотека с открытым исходным кодом; карты Гугл; технология Angular; обработка данных; Front-end приложение.

განხილვის თარიღი 27.09.2021

შემოსვლის თარიღი 29.09.2021

ხელმოწერილია დასაბეჭდად 28.12.2021